
NAMA	: MUHAMMAD FARREL YAFIWAGHANI FIRJATULLAH
NIM	: 22.01.53.0017
PRODI	: TEKNIK INFORMATIKA

WEB SERVICE

RANGKUMAN REQUEST METHOD HTTP.

1. GET

Metode GET digunakan untuk mengambil data dari server tanpa mengubahnya. Ini adalah metode umum yang digunakan ketika klien (seperti browser) meminta sumber daya dari server, misalnya halaman web atau data API. Misalnya, ketika Anda memasukkan URL di browser, itu mengirimkan permintaan GET ke server untuk mengambil konten. GET bersifat idempotent, artinya, memanggilnya berulang kali tidak akan mengubah data di server. Data biasanya dikirim melalui URL dan sering dapat di-cache untuk akses cepat di masa mendatang.

Fungsi Utama: Digunakan untuk mengambil atau membaca data dari server tanpa mengubahnya.

Contoh: Saat mengetik URL di browser dan menekan Enter, browser mengirimkan permintaan GET untuk mengambil konten dari URL tersebut.

Karakteristik:

1. **Idempotent:** Memanggil GET berkali-kali dengan request yang sama tidak mengubah hasil di server.
2. **Safe Method:** Tidak mengubah data di server, hanya mengambilnya.
3. **Cacheable:** Respons dari GET dapat di-cache oleh browser atau proxy.
4. **Request Body:** Tidak memiliki body; semua parameter dikirim melalui URL (query string).

2. POST

Metode POST digunakan untuk mengirimkan data ke server untuk membuat sumber daya baru. Data dikirim melalui body permintaan, bukan melalui URL seperti GET. POST sering digunakan untuk mengirimkan form atau mengunggah file. Tidak seperti GET, POST tidak idempotent, yang berarti menjalankan POST beberapa kali bisa menghasilkan hasil yang berbeda, seperti membuat sumber daya baru.

Fungsi Utama: Digunakan untuk mengirimkan data ke server dan membuat resource baru.

Contoh: Saat pengguna mendaftar di aplikasi, POST digunakan untuk mengirimkan data pengguna baru ke server.

Karakteristik:

1. **Non-idempotent:** Setiap kali POST dijalankan, dapat membuat resource baru yang berbeda.
2. **Create Operation:** Digunakan untuk pembuatan data baru.
3. **Request Body:** Data dikirim melalui body request, misalnya dalam format JSON atau form-encoded.
4. **Tidak Cacheable:** Biasanya tidak boleh di-cache karena mengubah data di server.

3. PUT

Metode PUT digunakan untuk memperbarui atau menggantikan sumber daya yang ada di server. Jika sumber daya sudah ada, ia akan diperbarui dengan data baru; jika tidak, sumber daya baru bisa dibuat. Metode ini juga bersifat idempotent, jadi memanggilnya berulang kali akan memberikan hasil yang sama. PUT biasanya digunakan untuk operasi penggantian penuh dari data yang ada.

Fungsi Utama: Digunakan untuk mengupdate resource yang sudah ada atau membuat resource baru jika belum ada.

Contoh: Pengguna memperbarui profilnya dengan mengganti data lama dengan yang baru melalui PUT.

Karakteristik:

1. **Idempotent:** Memanggil PUT berulang kali dengan data yang sama akan memberikan hasil yang sama.
2. **Replace Operation:** Menggantikan seluruh resource dengan data baru.
3. **Request Body:** Berisi data lengkap dari resource yang ingin diupdate.
4. **Tidak Cacheable:** Tidak boleh di-cache karena mengubah resource di server.

4. DELETE

Metode DELETE digunakan untuk menghapus sumber daya dari server. Setelah permintaan DELETE dikirimkan, sumber daya tersebut tidak lagi tersedia. Metode ini juga idempotent, karena memanggil DELETE beberapa kali pada sumber daya yang sama tidak akan mengubah hasilnya setelah dihapus.

Fungsi Utama: Digunakan untuk menghapus resource dari server.

Contoh: Pengguna menghapus akunnya dari aplikasi, server menerima request DELETE untuk menghapus data tersebut.

Karakteristik:

1. **Idempotent:** Menghapus resource yang sama beberapa kali tidak akan menghasilkan perubahan tambahan.
2. **Permanent Operation:** Menghapus data secara permanen (meskipun bisa diimplementasikan "soft delete").
3. **Tidak Cacheable:** Respons dari DELETE biasanya tidak di-cache.

5. OPTIONS

Metode OPTIONS digunakan untuk mengetahui metode HTTP apa saja yang diizinkan pada sumber daya tertentu. Ini sering digunakan untuk memeriksa kemampuan server atau dalam implementasi lintas domain (CORS), di mana browser memeriksa izin sebelum mengirimkan permintaan sebenarnya.

Fungsi Utama: Digunakan untuk mengetahui metode HTTP apa saja yang diizinkan di resource atau endpoint tertentu.

Contoh: Aplikasi client menggunakan OPTIONS untuk mengetahui apakah metode seperti GET, POST, atau PUT diizinkan di suatu resource.

Karakteristik:

1. **No Side Effects:** Tidak memodifikasi resource, hanya untuk komunikasi metadata antara client dan server.
2. **Used in CORS:** Sangat penting dalam Cross-Origin Resource Sharing (CORS) untuk memeriksa izin lintas domain sebelum request sebenarnya.

6. PATCH

Metode PATCH digunakan untuk memperbarui sebagian data dari sumber daya yang ada, berbeda dengan PUT yang menggantikan seluruh sumber daya. Metode ini cocok digunakan jika hanya sebagian kecil data yang ingin diubah. PATCH tidak selalu idempotent, karena perubahan bisa bersifat inkremental.

Fungsi Utama: Digunakan untuk mengupdate sebagian dari resource yang ada, bukan mengganti seluruh data seperti PUT.

Contoh: Pengguna mengupdate alamat emailnya tanpa mengganti informasi lain di profil.

Karakteristik:

1. **Non-idempotent:** Memanggil PATCH dua kali bisa memberikan hasil berbeda jika pengubahannya bersifat inkremental.
2. **Partial Update:** Hanya mengupdate sebagian resource.
3. **Request Body:** Hanya berisi bagian-bagian yang ingin diubah, tidak keseluruhan resource.

7. HEAD

Metode HEAD serupa dengan GET, tetapi hanya mengambil informasi header dari respon tanpa mengunduh isi atau kontennya. Ini berguna untuk memeriksa apakah sumber daya tersedia atau untuk melihat metadata sebelum mengambil data secara penuh dengan GET.

Fungsi Utama: Mirip dengan GET, tetapi hanya mengambil header dari respon tanpa mengambil body atau kontennya.

Contoh: Memeriksa ketersediaan halaman web tanpa mengunduh kontennya.

Karakteristik:

1. **Idempotent:** Seperti GET, tidak mengubah status resource.
2. **No Body:** Hanya mengambil header, tidak ada konten body.
3. **Used for Checking Availability:** Digunakan untuk memeriksa ketersediaan resource atau metadata tanpa mendownload konten.